Traffic Sensor Network

User's Manual

SD 1210

Group Members:

Bolan Gleason

Jesse Kohn

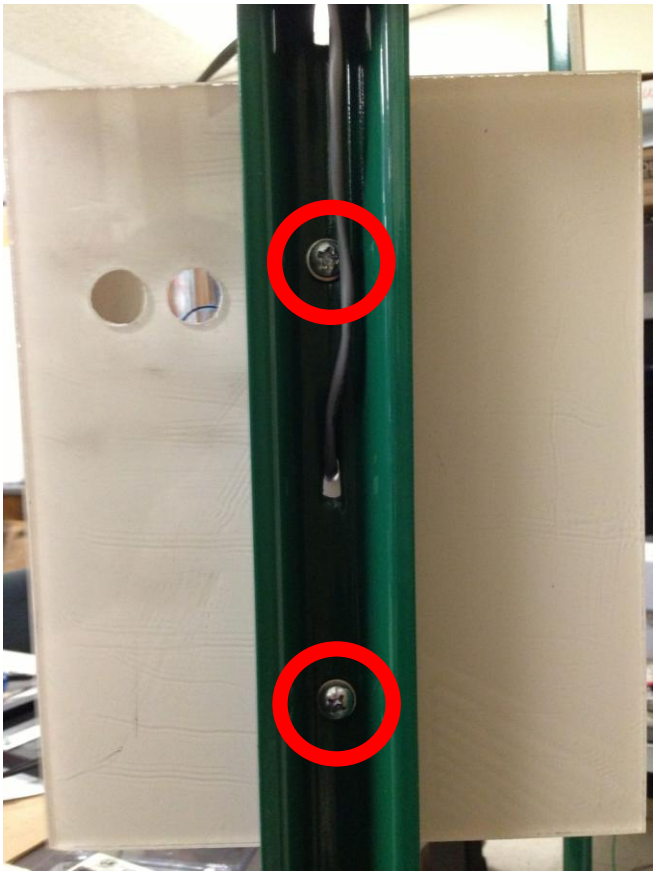
Zachary Leyk

Ian Wichmann

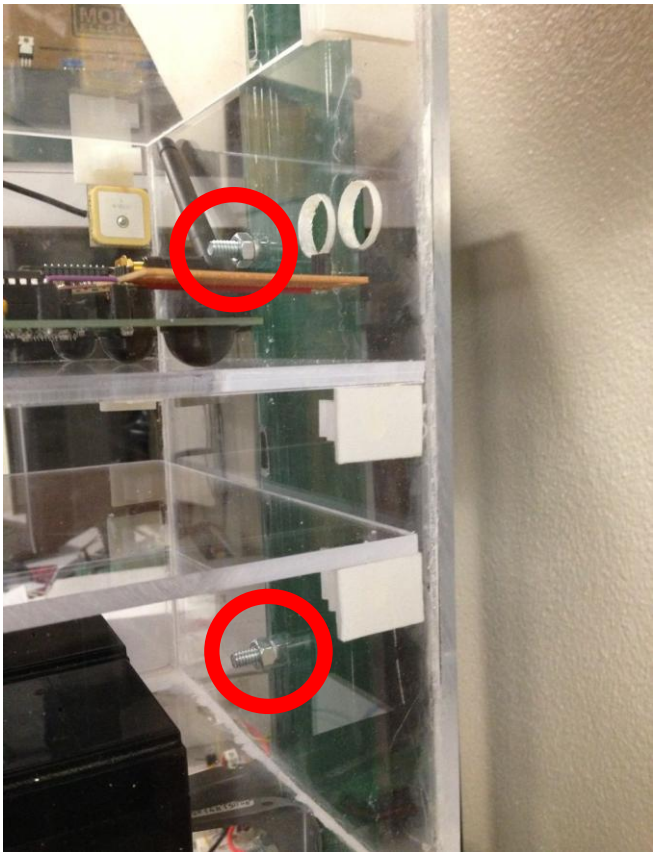
Mounting the Enclosure

To mount the enclosure and solar panel you will have to use your own discretion. For the sign we created, we use used the mounting points built into the enclosure box and an L-bracket underneath it to help hold it up. Then for the solar panel we had to create a mounting point on the back of the stop sign consisting of a small piece of wood bolted into the pole, which the panels mounting bracket could latch on to. The following pages show this in more detail.

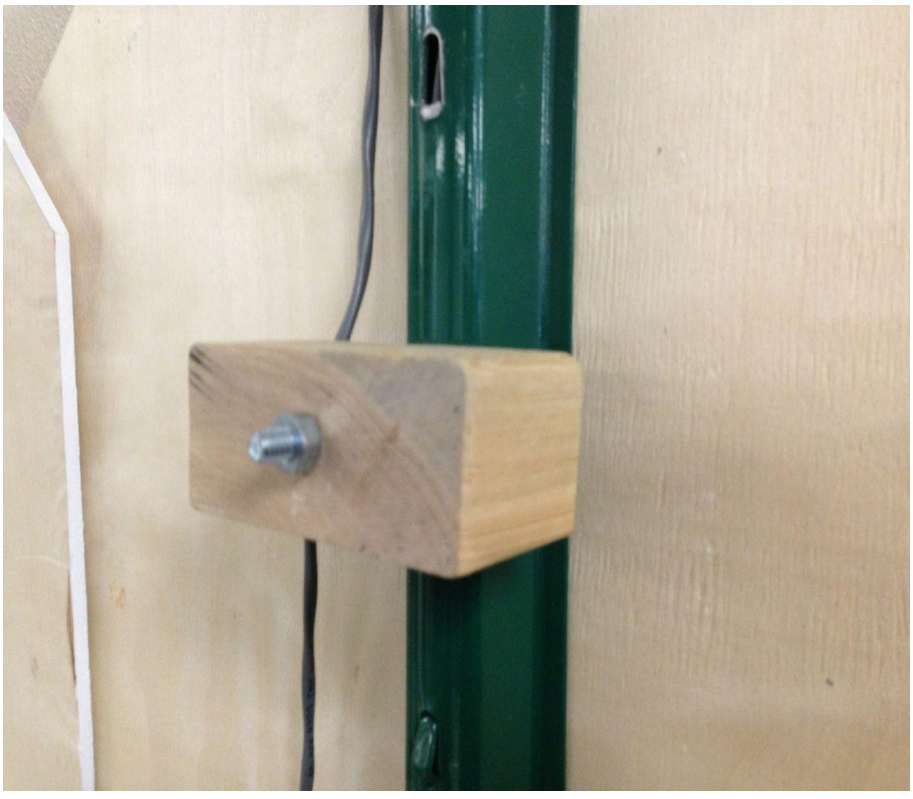
Here you can see the points where the L-bracket is mounted onto the sign post.



These are the mounting points on the front of the enclosure, where bolts go through the sign post and into the enclosure.



Our second enclosure used clear acrylic, so you can see the front mounting points here inside the enclosure.

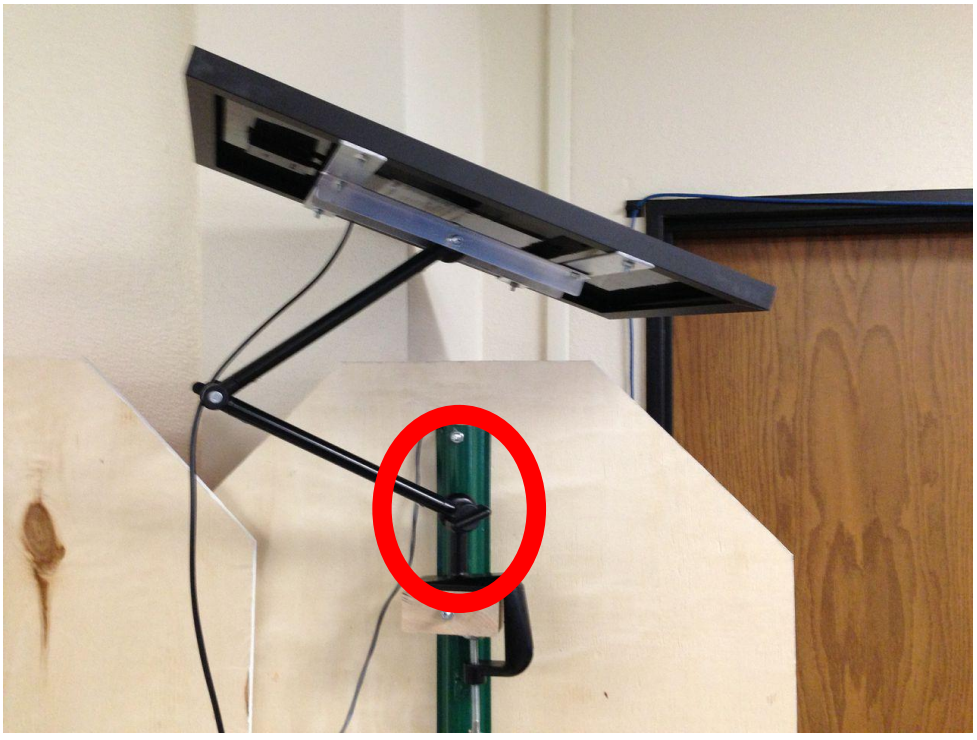


In order to mount the solar panel on the sign, we bolted this small piece of wood onto the back of the signpost.

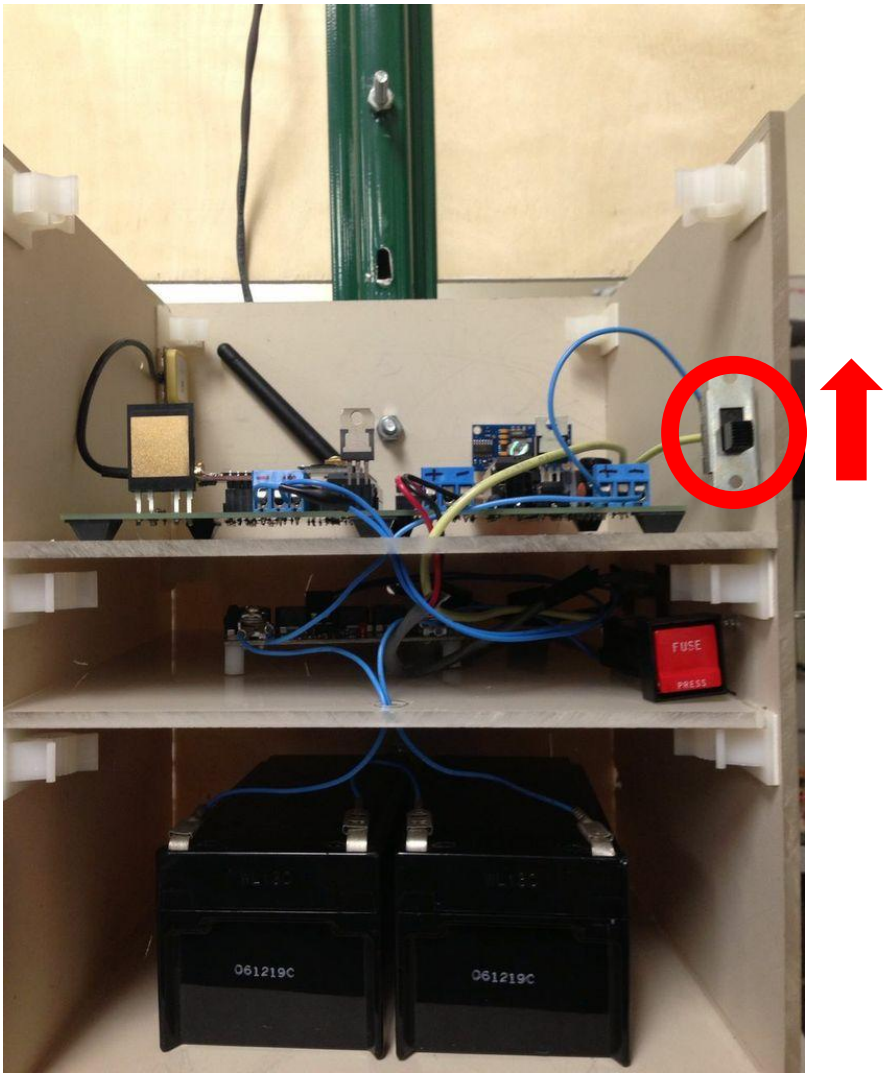
This is the mounting bracket for the solar panel. It tightens onto the wooden block on the back of the sign, and the solar panel mounting arm connects to the top part of the bracket that extends upwards.

Here you can see the bracket mounted onto the wooden block. You turn the arm on the bottom of the bracket to tighten the grip on the wooden block.

On top is the mounting arm of the solar panel, where you can see the post coming off the end that goes into the mounting bracket. The image below shows the solar panel connected to the mounting bracket.

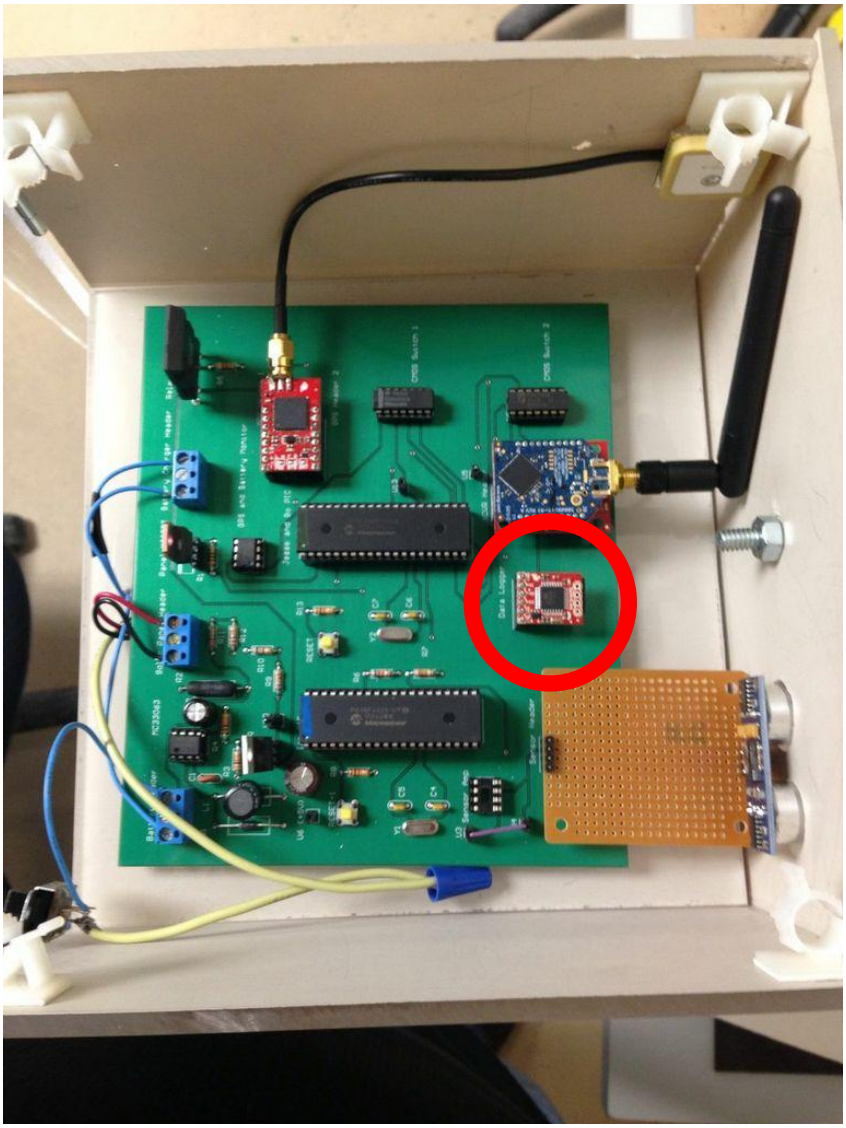


Activating the System



Once you have mounted the enclosure, to turn it on all you need to do is move the switch in the top right hand of the enclosure box to the upright position. To verify that the system is functioning you can observe several LEDs flashing on the top PCB.

Collecting Data



Once you have let the system run for some amount of time, you can collect the data. To do this, open the enclosure and remove the microSD card located underneath the data logger (circled in red). Then using a microSD to SD card adapter, plug the SD card into your PC. Then follow the instructions on the next page to run the Perl script that analyzes the data.

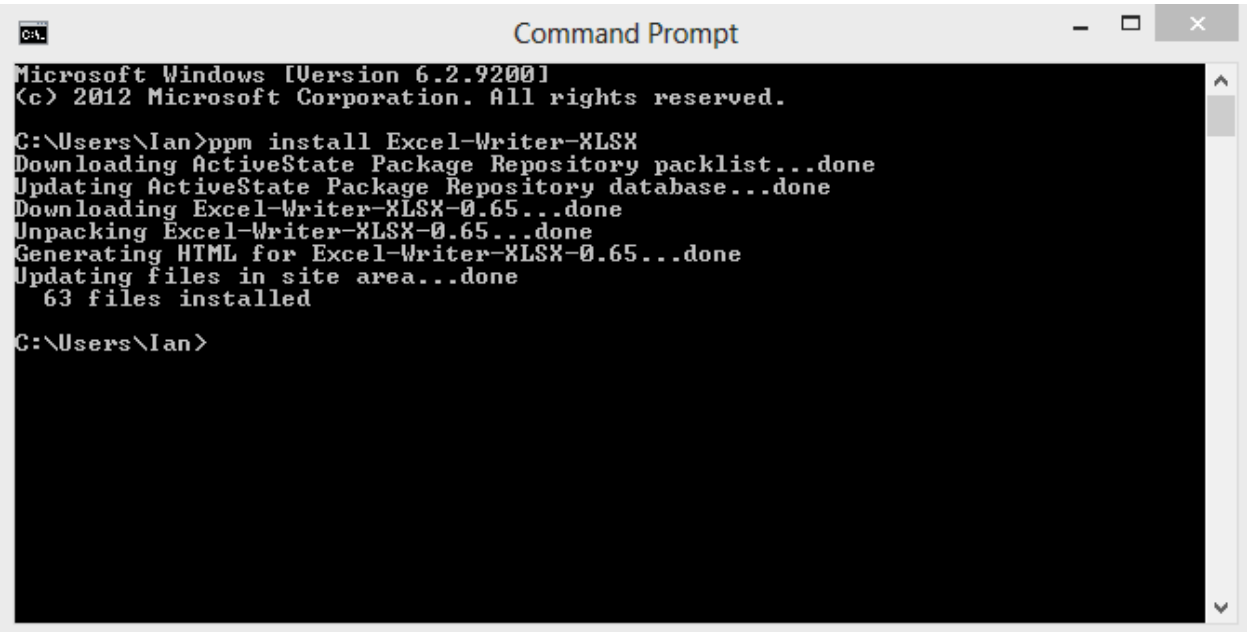
Running the Perl Script

The Perl script for our project parses all of the data saved by the data logger, exports the data into Microsoft Excel, and generates graphs of the data. In order to run the script you must have ActiveState Perl (5.16.3.1603+), the Excel::Writer::XLSX Perl package, and Microsoft Excel (2007+) installed on your PC. To install ActiveState Perl, you can go to this link and download the most recent version for free:

```
http://www.activestate.com/activeperl/downloads
```

To install the Excel::Writer::XLSX package open the command prompt utility by clicking WIN + R, typing cmd, and pressing enter. Type the following into the command prompt:

```
ppm install Excel-Writer-XLSX
```



Once you plug the SD card from the data logger into your PC, you will need to locate the file path of the data file you want to analyze. To do this, go to My Computer, and locate the drive path of the SD card. It will most likely be similar to this:

```
G:\ss1.txt
```

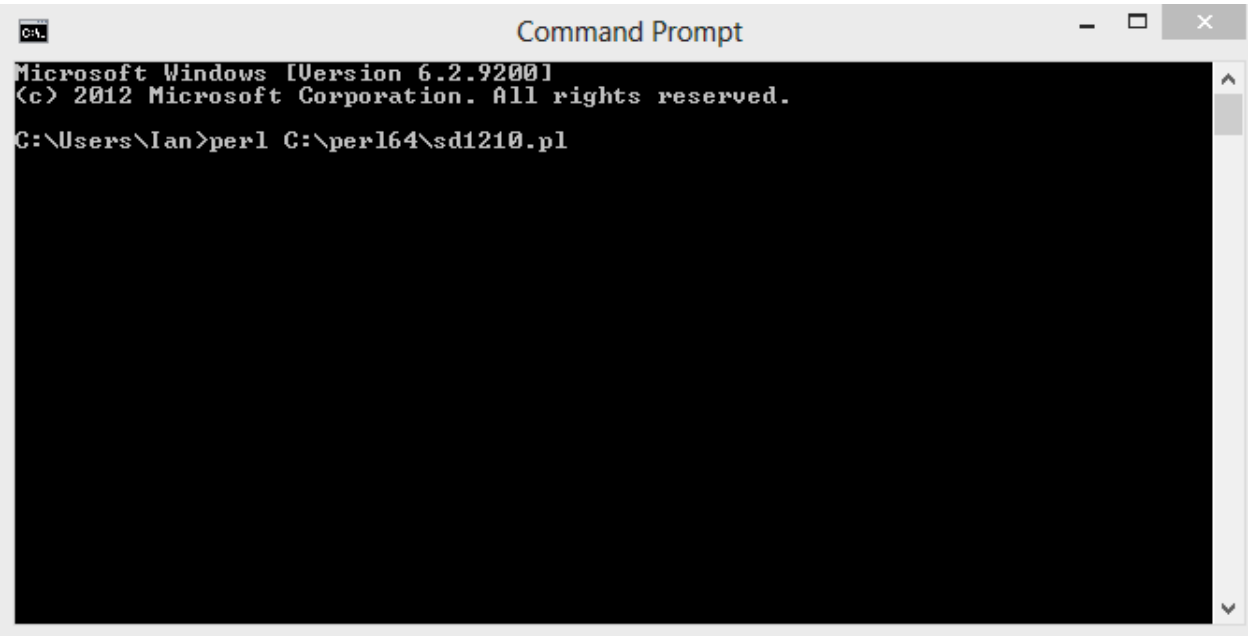
Then you must open the sd1210.pl file in a code editor (or notepad) and find line 34 and change ‘your file path here’ to the full path to the text file you found at the end of the previous page, which will look like:

$readFile = ‘your file path here’;

```
1  #================================================================================
2  # SD1210 - Result Graphing Script - PERL
3  # Written by: Ian Wichmann
4  # Date: 4/12/2103
5  #
6  # Objective: Extract data from .txt file, organize it, and plot it
7  #               using Excel::Writer::XLSX
8  #  Data Format: SS1DDMMYYHRMNSSssss
9  #================================================================================
10 #!/usr/bin/perl -w
11
12 use strict;
13 use Excel::Writer::XLSX;
14
15 my $elapsedHours = 0;
16 my $measurementsInHour;
17 my $averageSpeed;
18 my $evalHour;
19 my $hoursIndex = 1;
20 my $fileLength;
21 my $readFile;
22 my $carsIndex = 0;
23 my @hours;
24 my @linesInHour;
25 my @hourlyAverageSpeed;
26 my @averageSpeedByHour;
27 my @hours;
28 my @fileLines;
29 my @carsPerHour;
30
31 my $workbook = Excel::Writer::XLSX -> new('sd1210.xls');
32 my $worksheet = $workbook -> add_worksheet();
33
34 $readFile = 'G:\ss1.txt';
35 open FILE, '<', $readFile or die "$!";
36 @fileLines = <FILE>;
37 $fileLength = @fileLines;
```

Note: this step is very important because if you do not enter the file location into the Perl script correctly the script will not be able to run.

To run the Perl script, go to the command prompt, then type in the following line (note that the file path must point to where you stored the sd1210.pl file, which may not be the same as what is in the example below):

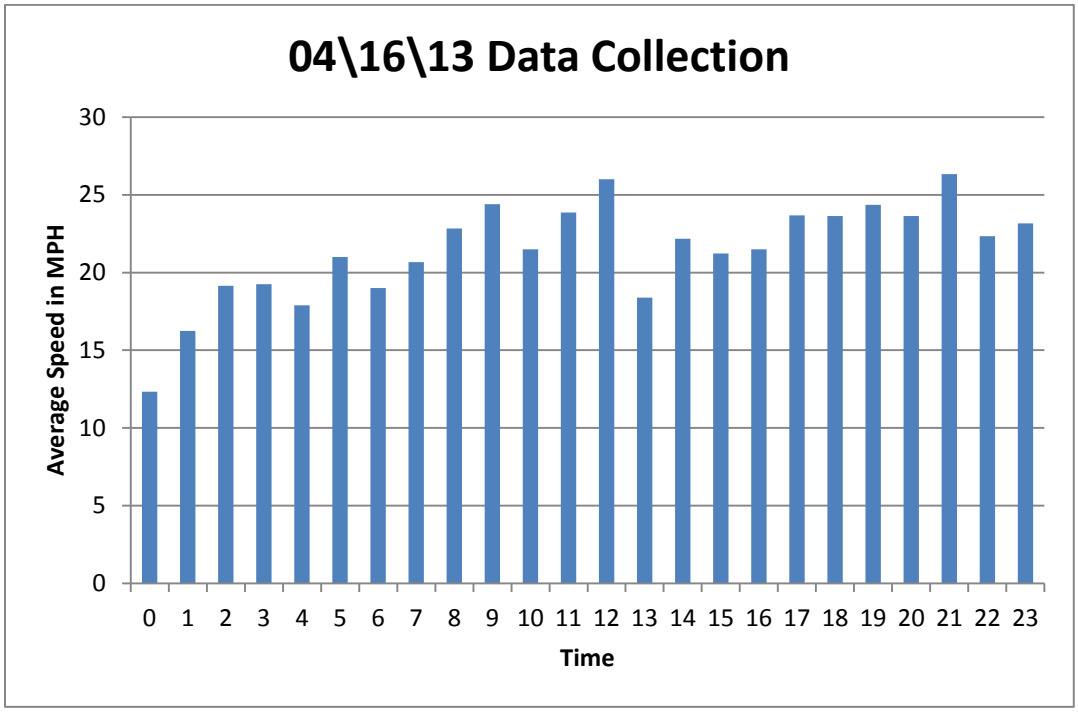


You should be able to find the script in your My Documents folder, if it is not present there do a Windows search for sd1210.xls. The data will be displayed in the first three sheets in Excel. The first sheet will display the analyzed time (24 hour format) with the average speed and number of cars recorded in that hour. The second and third sheets are graphs of the values displayed on the first sheet.

Sheet 1

	Time	Average Speed in MPH		Time	Number of Cars
2	0	12.33333333		0	6
3	1	16.25		1	4
4	2	19.14285714		2	7
5	3	19.25		3	4
6	4	17.88888889		4	9
7	5	21		5	3
8	6	19		6	9
9	7	20.66666667		7	3
10	8	22.83333333		8	6
11	9	24.4		9	5
12	10	21.5		10	4
13	11	23.85714286		11	7
14	12	26		12	3
15	13	18.375		13	8
16	14	22.16666667		14	6
17	15	21.22222222		15	9
18	16	21.5		16	12
19	17	23.66666667		17	6
20	18	23.625		18	8
21	19	24.36363636		19	11
22	20	23.625		20	16
23	21	26.33333333		21	6
24	22	22.33333333		22	9
25	23	23.16666667		23	6

Sheet 2



Sheet 3